9. Real-Time Services

9.1 Overview

The real-time services discussed in this section include: logical string processing, Telemetry Service, commanding service, and ground telemetry processing. Logical string processing services provide FOT and IST users the ability to employ real-time services in order to monitor and control real-time, simulation, and historical replay activities for a given spacecraft. The Telemetry Service provides the ability to monitor spacecraft telemetry associated with real-time, simulation, and historical replay activities. The Command Service provides the ability to send scheduled and real-time commands to the spacecraft as part of a real-time logical string, and to a simulated spacecraft as part of a simulation logical string. The Ground Telemetry Processing Service provides the ability to monitor the Network Control Center (NCC) and EDOS external interfaces as part of a real-time logical string. Historical replay capabilities are also provided using archived external interface data as part of a replay logical string.

Real-time services are driven by ECL directives. ECL directives invoking and customizing real-time services can be manually entered via the command line on the Control window or Command Control window. Alternatively, real-time ECL directives can be entered into the system through automated means, such as command procedures, ground procedures, and ground scripts. For a discussion of the various mechanisms by which directives are entered into the system, refer to Section 6. This section provides the following information for each ECL directive used to invoke real-time services:

a. A description of the keywords associated with a directive.

b. Valid mechanisms for entering a directive into the system.

c. Event messages that can be expected after entering a directive.

d. The database version associated with the logical string (i.e., 1.0).

Additional information regarding ECL directives, their usage, event definitions, and recommended responses may be found in Appendices A and B.

9.2 Logical String Processing

A logical string can be defined as the hardware hosts and the application software supporting a real-time, simulation or historical replay activity for a given spacecraft. An activity in this context can be described in terms of three attributes that collectively and uniquely identify each logical string established by the system. The three attributes include:

a. An identifier for the specific EOS spacecraft being supported (e.g., AM-1).

b. The source of the data being processed (i.e., real-time, simulation, replay).

c. The intended use of the logical string (i.e., operational, test, training).

Logical strings employ shared or dedicated resources and are either globally visible or visible only on the userstation establishing the string. A shared logical string is established with an EOC Real-Time server as its host and is globally visible to users within the EOC and users connected to the EOC through ISTs. A dedicated logical string is established on a userstation at the EOC or at an IST, and is dedicated to and accessible by a single user. Some capabilities accessed via connection to a shared logical string include:

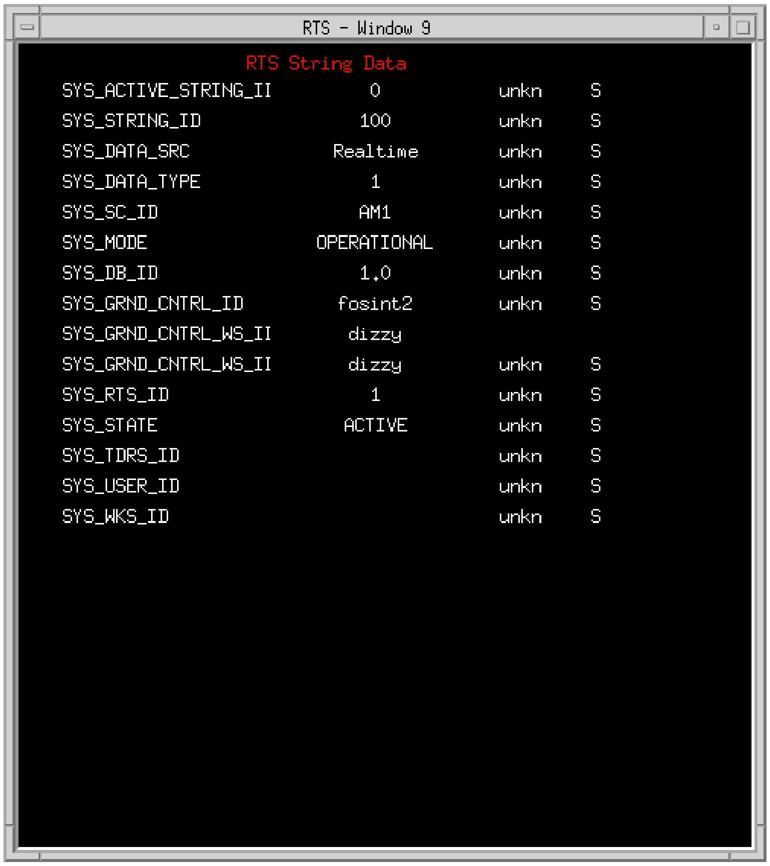
a. Operational monitoring of a real-time contact with a given EOS spacecraft.

b. Simulation of real-time contacts with a given EOS spacecraft for training purposes.

c Replay of historical spacecraft telemetry from a given EOS spacecraft to support the testing of new release software.

Shared logical strings are created to support real-time, simulated, and historical replay activities because they concurrently provide multiple EOC and IST users access to the same data streams. Only users at the EOC with ground control privileges may establish shared logical strings. Dedicated logical strings support off-line activities, such as historical replays and analysis requests and may be created from an IST or from the EOC by any user.

Logical strings are identified by a numeric identifier ranging from 1 to 999. The identifier for a shared logical string is unique; for example, shared logical string 100 will be the same on every userstation within the EOC as well as every userstation at an IST. A dedicated logical string is identified by a numeric identifier less than 100. Since dedicated logical strings are visible only on the local userstation, logical string 002 on one userstation is not related in any way to logical string 002 on another userstation. Figure 9.2-1 illustrates a sample dynamic page containing logical string configuration parameters.



```
                    RTS String Data
SYS_ACTIVE_STRING_II          0          unkn    S
SYS_STRING_ID               100          unkn    S
SYS_DATA_SRC             Realtime        unkn    S
SYS_DATA_TYPE                 1          unkn    S
SYS_SC_ID                   AM1          unkn    S
SYS_MODE              OPERATIONAL        unkn    S
SYS_DB_ID                   1.0          unkn    S
SYS_GRND_CNTRL_ID        fosint2         unkn    S
SYS_GRND_CNTRL_WS_II      dizzy
SYS_GRND_CNTRL_WS_II      dizzy          unkn    S
SYS_RTS_ID                    1          unkn    S
SYS_STATE                ACTIVE          unkn    S
SYS_TDRS_ID                              unkn    S
SYS_USER_ID                              unkn    S
SYS_WKS_ID                               unkn    S
```

Figure 9.2-1. Logical String Configuration Display

Logical string 100 in Figure 9.2-1 is an example of a real-time operational logical string. This type of string is unique in that it consists of two strands. One strand of software processes supports the operational spacecraft mission and provides the user access to real-time telemetry data and other command and control capabilities. The second strand represents the backup capability in place in the event of a hardware or software failure that renders the active strand inoperable. The significance of this is further described in paragraph 9.5.

9.2.1 Logical String Processing Directives

ECL directives for real-time logical string processing are derived from either the STRING or TAKE primary keyword. Off-line analysis requests result in the creation of a logical string for historical telemetry processing without the STRING or TAKE directive. For these requests, the directive acting as the catalyst is issued by the Analysis Request Builder. Additional detail concerning the Analysis Request Builder is available in paragraph 10.2.1.

The STRING directive has associated secondary keywords for creation and configuration of a logical string, connection to an existing logical string, and transfer of control of mission-critical processing from a string's active processing to its backup processing. The TAKE directive has two associated secondary keywords specifying the privileges that will be requested to perform mission-critical functions within the EOC.

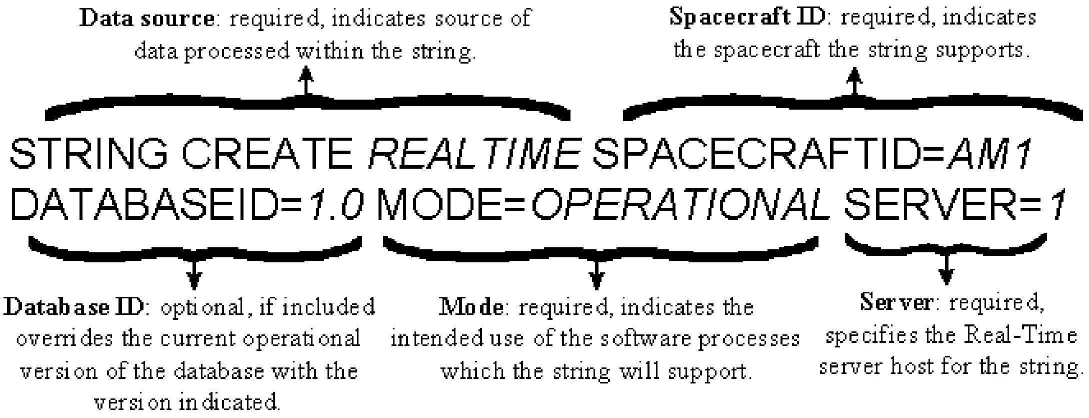
9.2.2 Logical String Creation

Logical strings can be established manually through a directive or by default when the Real-Time server startup script is run. Nominally, logical strings are established manually. Default logical string creation is used only in conjunction with carefully established and maintained operational procedures. The following sections discuss manual and default logical string creation in detail.

9.2.2.1 Manual Logical String Creation

Logical strings are established manually by issuing the STRING CREATE directive from the Control window ECL command line or by requesting the off-line analysis of historical telemetry described in paragraph 10.2. The spacecraft, database ID, data type, replay type (dedicated or shared), replay rate, start time and stop time for the replay are entered via the Replay Control window. The logical string for processing historical telemetry is generated through the submission of an off-line analysis request via the Data Replay Control window.

The STRING CREATE directive requires additional secondary directives that specify the configuration of the software executed as part of the logical string. These directives designate the data source, spacecraft ID, database ID, mode and Real-Time server for the string and are explained in detail in the following text. Figure 9.2.2.1-1 provides a sample STRING CREATE directive with associated subdirectives.



Data source: required, indicates source of data processed within the string.

Spacecraft ID: required, indicates the spacecraft the string supports.

```
STRING CREATE REALTIME SPACECRAFTID=AM1
DATABASEID=1.0 MODE=OPERATIONAL SERVER=1
```

Database ID: optional, if included overrides the current operational version of the database with the version indicated.

Mode: required, indicates the intended use of the software processes which the string will support.

Server: required, specifies the Real-Time server host for the string.

Figure 9.2.2.1-1. Sample* STRING CREATE *Directive

a. **Data Source**. The type of data being processed within a logical string is referred to as the data source and is required in the STRING CREATE directive. Data processed within a logical string can be from one of three sources: an EOS spacecraft, an EOS spacecraft simulator, or a historical archive. Logical strings that process EOS spacecraft or EOS spacecraft simulator data are established through the STRING CREATE directive using the *REALTIME* or *SIMULATION* data source keywords, respectively.

b. **Spacecraft ID**. The EOS spacecraft ID is required in the STRING CREATE directive. Although the need to specify the spacecraft identifier may seem obvious, the spacecraft identifier is required because logical strings are key components in the simultaneous support of multiple EOS missions.

c. **Database ID**. The DATABASEID keyword and value is used to override the current operational version of the project database and is optional in the STRING CREATE directive. When the database ID is not specified in the directive, the system will establish the logical string using the latest available version of the project database.

d. **Mode**. The mode defines the intended use of software executed by the logical string request and is required in the STRING CREATE directive. Valid logical string modes include: operational, training, and test. The operational mode is specified when the logical string will directly support control center operations. Critical activities such as spacecraft commanding and ground system configuration and control can only be performed through a connection to a logical string which has a real-time data source and an operational mode.

e. **Real-Time server**. The Real-Time server identifier is required information for the STRING CREATE directive. The server keyword allows you to specify the Real-Time server host for the desired logical string. This flexibility enables you to consider the resource loading on each available server when creating a new logical string. In order for backup strings to serve their intended purpose for failure recovery, backup processing for a real-time string must be on a different Real-Time server than the active processing for a corresponding real-time string.

To manually create a real-time, operational logical string via the STRING CREATE directive:

1. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying logical string configuration information.

2. **Create a logical string**.

   Enter the STRING CREATE directive in the Control window command line:

   STRING CREATE *REALTIME* SPACECRAFTID=*AM1* DATABASEID=*1.0*

   MODE=*OPERATIONAL* SERVER=*1*

3. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the string creation. Event messages are generated for key software processes spawned during string creation. If a problem occurs during string creation, any processes that were created will be brought down. For each terminated process, an event message will be generated. Table 9.2.2.1-1 identifies the text of event messages associated with string creation or termination and their significance.

Table 9.2.2.1-1. String Creation/Termination Event Message Text

Event Message Text	Significance
Establishing __ service	Initialization of a process.
__ process __ configured	The string creation process has been initialized.
String __ was __	The string has been created.
Shutting down __ process	The process is being shut down.

4. **Monitor the parameter updates on the dynamic page**.

5. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Global Event Display** from the list of tools and click **OK**.

6. **Connect to a logical string**.

   You must connect to a string in a mirrored configuration before assuming ground control. In the Command window’s command line enter:

   STRING CONNECT STRING=*100* TLMTYPE=*HOUSEKEEPING*

   CONFIG=*MIRROR*

7. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is not successful, investigate why by reviewing the event messages generated and repeat step 6 before proceeding.

8. **Assume ground control privilege**.

   Enter the TAKE GROUND CONTROL directive in the command line of the Command Control window:

   TAKE GROUNDCONTROL STRING=*100*

9. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the request for ground control privilege. If ground control privilege is granted, an event message stating "Ground Control Authority has changed from__to__" will be generated.

9.2.2.2 Backup Logical Strings

In order to facilitate the most expedient and efficient recovery possible from a system hardware or software component failure, the system uses redundant processing executing on separate Real-Time servers. This type of processing operates as a "hot backup" for mission-critical processing provided via logical stings with a real-time data source and operational mode.

In order to take advantage of this redundant processing, a backup logical string must be established for any logical string with a real-time data source and operational mode. Backup logical strings are established manually by an authorized member of the FOT by appending the BACKUP keyword to the STRING CREATE directive:

STRING CREATE *BACKUP* SPACECRAFTID=*AM1* DATABASEID=*1.0* MODE=

OPERATIONAL SERVER=*2*

9.2.2.3 Simulation Logical Strings

The STRING CREATE SIMULATION directive establishes a logical string capable of interacting with a spacecraft simulator. A simulation string is very similar to a real-time string except for the number of processes spawned and the data source of the logical string. Fewer processes may be spawned because not all real-time processes are needed to communicate with a spacecraft simulator. For example, Real-Time Contact Management software is used to communicate with EDOS and NCC for AM-1, not the spacecraft simulator. Therefore, it is not spawned for a simulation string.

9.3 Default Logical String Creation

Default logical strings are created by the system when the Real-Time server startup script is run. Operationally, the default logical string creation capability can be used to ensure that a logical string with a real-time data source and operational mode is started as soon as the Real-Time server software is started. The full path name for the file read by the system to create a particular default logical string is:

/fos/release/am1/odf/DefStringOdf_1.0

where:

/fos/<release>/ <spacecraft>/DefStringOdf_1.0

<release> = fosa, fosb, etc.

<spacecraft> = am1, pm1, etc.

To disable the creation of a default logical string at Real-Time server startup, the DefStringOdf_1.0 file created during ODF generation can be deleted or replaced with an empty file by the same name. The system will recognize that the file is absent or empty at startup and will not create a logical string. The intended use of this capability will be fully described in operational procedures developed by the ECS FOT.

9.4 Failure Detection

Mission-critical flight operations support hardware and software components are automatically monitored. Status parameters of monitored components may be displayed on a dynamic page. In the event of failure of a real-time operational string component, the following messages detailed in Table 9.4-1 are generated and displayed on the Global Event Display window to guide you in proceeding with failure recovery.

Table 9.4-1. Failure Recovery Event Message Text

Event Message Text	Significance
Active strand for string ID ____________ failed due to failure of process. Suggest failover	The active strand for the string indicated has failed. Perform failover to the backup strand.
Backup strand for string ID ___________ failed due to failure of process. Suggest string delete BACKUP strand	The backup strand for the string indicated has failed. Delete the backup strand.
Realtime server__________, host name__________, failed.	The Real-Time server host indicated failed.
Complement for local string _________unavailable. _________string assigned to failed realtime server _______.	The complement for the local string indicated is unavailable.
Software monitor request failed for string __________. Suggest delete string.	The software monitor request for the string indicated failed. Delete the string.
Realtime server __________, host name _________ , available.	The Real-Time server host indicated is available.

9.5 Failure Recovery

The following sections describe failure recovery procedures recommended in the event of hardware or software component failures affecting mission-critical flight operations support. Software executed as part of a real-time, operational logical string in the active state is considered mission-critical. All other types of logical strings executing within the system are for off-line analysis, software testing, or personnel training and are not deemed critical to the life of an EOS spacecraft.

A real-time, operational string is composed of an active and backup strand. The active strand is created manually via the STRING CREATE directive or through the DefStringOdf_1.0 file (refer to paragraph 9.2.2.1). Once the active strand is created, the backup strand is created by an authorized member of the FOT via the STRING CREATE BACKUP directive. The backup strand mirrors the active processing of the real-time, operational string and ensures that if a Real-Time server fails within the EOC, or if a critical software process executing in an active, mission-critical logical string fails, an operator will be able to recover from the failure in the shortest possible time. Failure to establish backup processing for each real-time, operational string may prevent operations personnel from recovering from a hardware or software failure within the one minute system requirement.

As long as a backup string has been created for the real-time operational string, string failover may be performed via the STRING FAILOVER directive which transfers control of the spacecraft from the active to the backup string. The Ground Controller for a real-time, operational string is the only individual authorized to perform a string failover. The only additional information required for the STRING FAILOVER directive is the STRING ID identifying the failed string.

To perform failover for a real-time operational logical string:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Global Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   You must connect to a string in a mirrored configuration before assuming ground control. In the Control window's command line enter:

   STRING CONNECT STRING=*100* TLMTYPE=*HOUSEKEEPING* CONFIG=
   MIRROR

3. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is not successful, investigate the reason why by reviewing the event messages generated and repeat step 2 before proceeding.

4. **Assume ground control privilege**.

   Enter the TAKE GROUND CONTROL directive in the command line of the Command Control window:

   TAKE GROUNDCONTROL STRING=*100*

5. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the request for ground control privilege. If ground control privilege is granted, an event message stating "Ground Control Authority has changed from__to__" will be generated.

6. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying logical string configuration information.

7. **Enter the STRING FAILOVER directive in the Control window command line**.

   STRING FAILOVER STRING=*100*

8. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the transfer of control from the active to the backup string. During a failover, event messages are generated as processes are deactivated and as the backup processes are activated. Tables 9.5-1 and 9.5-2 identify the text of event messages associated with successful and unsuccessful string failovers and their significance.

Table 9.5-1. Successful String Failover Event Message Text

Event Message Text	Significance
Shutting down __ process.	Processes are being terminated.
Successfully terminated __ Process ID __.	The process has been successfully terminated.

Table 9.5-2. Unsuccessful String Failover Event Message Text

Event Message Text	Significance
Unable to change state of __ process to __.	A problem has occurred during activation or deactivation of a process.
Unsuccessfully terminated __ Process ID __.	The shutdown was unsuccessful.

If the shutdown is unsuccessful, enter the STRING DELETE directive in the command line of the Control window to terminate the inactive processing as described in paragraph 9.8 and create a new string using the STRING CREATE directive.

9. **Monitor the parameter updates on the string configuration dynamic page**.

   Two key parameters indicate the successful completion of the failover. First, the RTS ID parameter will be changed to the value previously associated with the backup RTS ID parameter. Second, the backup RTS ID parameter will be set to 0. If these values are not set properly, the failover was unsuccessful. Delete the existing string and create a new string as described in step 8.

9.6 Connect to a Logical String

The STRING CONNECT directive associates a userstation with an established logical string. The following instructions explain how EOC and IST users determine which logical strings have been established at the EOC and are available for connection.

To determine which logical strings are available for connection:

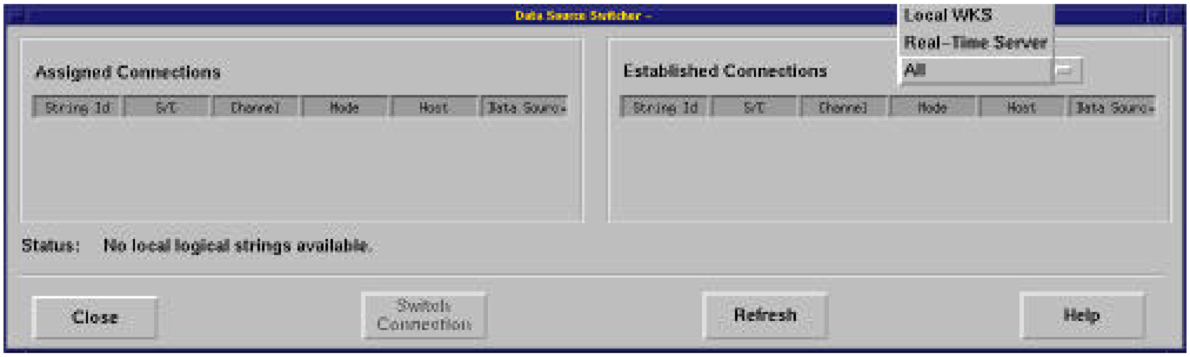
1. **Open a dynamic page**.

   Click **TlmWins...** on the Control window to open the Dynamic Page dialog box. Select a dynamic page from the list and click **OK**.

2. **Open the Data Source Switcher window (see Figure 9.6-1)**.

   Position the pointer on the dynamic page, click the right mouse button and select **Data Source Switcher** from the menu.

The Data Source Switcher window displays all logical strings you are connected to in the Assigned Connections table. All logical strings available for connection, including strings you are connected to, are listed in the Established Connections table. Choose one of the three options from the pull-down menu above the Established Connections table to display strings with processes running on your userstation (**Local WKS**), strings with processes running on the Real-Time server (**Real-Time Server**), or all logical strings (**All**).



Status: No local logical strings available.

Figure 9.6-1 . Data Source Switcher

Logical strings with identifiers between and including 100 and 200 are real-time strings available for connection. Logical strings with identifiers less than 100 represent dedicated historical replay or off-line analysis activities executing locally on the userstation. These strings are generated as a by-product of historical replay or off-line analysis and the connection to the string is accomplished by the software rather than via the STRING CONNECT directive.

The request for connection to a logical string via the CONNECT directive implies that resources on the local userstation may be employed to provide the access to the requested activity. Specify the numeric identifier for the string with which a connection is desired as well as the type of connection. There are two connection types available, mirrored and tailored. A mirrored connection inherits the configuration defined by the Ground Controller for that particular logical string. A mirrored connection is requested by specifying a mirrored configuration in the STRING CONNECT directive, such as STRING CONNECT STRING=*100* CONFIG=*MIRROR*.

A tailored connection allows you to tailor the configuration of local resources to specifications that differ from those of the Ground Controller. Upon connection, tailored users inherit the configuration of the Ground Controller. However, once the initial connection is made, the configuration of local resources may be altered and is unaffected by ground configuration changes made by the Ground Controller. A tailored connection is requested by specifying a tailored configuration in the STRING CONNECT directive, such as STRING CONNECT STRING=*100* CONFIG=*TAILOR*.

In order to assume the Ground Control privilege and modify the ground configuration for all mirrored users, establish a connection to the logical string with a mirrored configuration. If you connect to a logical string in a tailored configuration, you may not receive the Ground Control privilege in order to affect the configuration established for other users connected to the string.

To request a connection to a logical string:

1. **Open the Local Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   In the Control window command line enter the STRING CONNECT directive to establish a connection to the logical string in a mirrored or tailored configuration:

   Connect to a string in a mirrored configuration:

   STRING CONNECT STRING=*100* CONFIG=*MIRROR*

   or

   Connect to a string in a tailored configuration:

   STRING CONNECT STRING=*100* CONFIG=*TAILOR*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying user connection information.

4. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the connection to the string. During string connection, event messages are generated as processes are initialized. If an error occurs during string connection, any process spawned as a result of the string connection will be terminated and event messages reflecting the shutdown of processes will be generated. If the STRING CONECT directive is unsuccessful, investigate why and repeat step 2. Table 9.6-1 identifies the text of event messages associated with string connection and their significance:

Table 9.6-1. String Connection Event Message Text

Event Message Text	Significance
Establishing __ service.	Initialization of a process.
__ process __ configured.	The string creation process has been initialized.
__ connected to String __.	Successfully connected to the string.
Shutting down __ process.	An error was encountered during the string connection. A process is being shut down.

5. **Monitor the parameter updates on the dynamic page**.

   Once the connection is established, your login name will be added to the list of users connected to the string. Your userstation will be added to either the mirrored userstation list or the tailored userstation list, depending on the configuration of your connection.

9.7 User Authorization

There are two privileges administered by the real-time software subsystems: command authority and ground control privilege. EOC operators must acquire command authority before sending real-time commands to an EOS spacecraft, or initiating communication through the Ground Telemetry Processing Service with the NCC. Similarly, EOC operators must acquire Ground Control privilege before modifying the ground configuration of the system. In this context, ground configuration is the allocation or modification of any ground system resource within the EOC.

There are two configuration files for each privilege read by the application software at request time. The first is used to determine whether the requester is eligible to receive the requested privilege. The second is used to determine if the requested privilege can be acquired from the userstation where the user is logged on. The following sections describe how the command authority and ground control privileges are managed by the system and requested by users.

9.7.1 Command Authority

In order to be eligible to receive command authority, your login ID must be on the command authority user list accessed at request time by the system. The EOC System Administrator maintains the command authority user list based on operational procedures to be established by the FOT. The full path name for the command authority user file is:

/fos/release/am1/config/user.db

where:

/fos/<release>/ <spacecraft>/config/user.db

<release> = fosa, fosb, etc.

<spacecraft> = am1, pm1, etc.

Userstations from which you can request and receive command authority must be listed by host name on the command userstation list accessed at request time by the system. The EOC System Administrator maintains the command userstation list based on operational procedures to be established by the FOT. The full path name for the command authority user file is:

/fos/release/am1/config/hw.db

where:

/fos/<release>/ <spacecraft>/config/hw.db

<release> = fosa, fosb, etc.

<spacecraft> = am1, pm1, etc.

To request command authority:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   In the Control window's command line enter the STRING CONNECT directive to establish a connection to the logical string in a mirrored configuration:

   STRING CONNECT STRING=*100* CONFIG=*MIRROR*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying the current command user login identifier and command userstation identifier.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is not successful, investigate why by reviewing the event messages generated and repeat step 3 before proceeding.

5. **Assume command authority**.

   Enter the TAKE COMMAND directive in the command line of the Command Control window:

   TAKE COMMAND STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify the progress of the authority request. If command authority is granted, an event message stating "Command Authority has changed from __ to __" will be generated.

7. **Monitor the parameter updates on the dynamic page**.

   Observe updates to the current command user login identifier and command userstation identifier.

9.7.2 Ground Control Privilege

In order to be eligible to receive ground control privilege, your login ID must be on the ground control user authorization list accessed at request time. The EOC System Administrator maintains the ground control user authorization list based on operational procedures to be established by the FOT. The full path name for the file is:

/fos/release/am1/config/gcuser.db

where:

/fos/<release>/ <spacecraft>/config/gcuser.db

<release> = fosa, fosb, etc.

<spacecraft> = am1, pm1, etc.

Userstations where you can request and receive the ground control privilege must be listed by host name on the ground control userstation list accessed at request time. The EOC System Administrator maintains the ground control userstation list based on operational procedures to be established by the FOT. The full path name for the ground control userstation file is:

/fos/release/am1/config/gchw.db

where:

/fos/<release>/ <spacecraft>/config/gchw.db

<release> = fosa, fosb, etc.

<spacecraft> = am1, pm1, etc.

To request the ground control privilege:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   You must connect to a string in a mirrored configuration before assuming ground control. In the Control window's command line enter:

   STRING CONNECT STRING=*100* TLMTYPE=*HOUSEKEEPING* CONFIG=*MIRROR*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying the current ground control user login identifier and ground control userstation identifier.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is not successful, investigate the reason why by reviewing the event messages generated and repeat step 3 before proceeding.

5. **Assume ground control privilege**.

   Enter the TAKE GROUND CONTROL directive in the command line of the Command Control window:

   TAKE GROUNDCONTROL STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the request for ground control privilege. If ground control privilege is granted, an event message stating "Ground Control Authority has changed from__to__" will be generated.

7. **Monitor the parameter updates on the dynamic page**.

   Observe parameter updates to the current ground control user login identifier and ground control userstation identifier.

9.8 Logical String Deletion

Logical strings are deleted via a STRING DELETE directive specifying the string's numeric identifier, or upon completion of historical telemetry off-line analysis. Shared logical strings can only be deleted by the individual with ground control privilege for the string, after all users have disconnected from the string. In other words, to delete a shared string you connect to the string, assume ground control privilege for the string, disconnect from the string, and then delete the string as described in the following text.

For real-time, operational logical strings, the directive should specify the state of the strand to be deleted, active or backup. If no state is entered, the active strand will be deleted by default. The active strand can only be deleted after all backup processing that exists for the string has been deleted. Backup processing is deleted by deleting the backup strand.

Dedicated strings may be deleted by entering the STRING DELETE directive on the userstation where the string resides.

To delete a shared logical string:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   You must connect to a string in a mirrored configuration before assuming ground control. In the Control window's command line enter:

   STRING CONNECT STRING=*100* TLMTYPE=*HOUSEKEEPING* CONFIG= *MIRROR*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying all available logical strings.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is not successful, investigate why by reviewing the event messages generated and repeat step 3 before proceeding.

5. **Assume ground control privilege**.

   Enter the TAKE GROUND CONTROL directive in the command line of the Command Control window:

   TAKE GROUNDCONTROL STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe the progress of the request for ground control privilege. If ground control privilege is granted, an event message stating "Ground Control Authority has changed from__to__" will be generated.

7. **Disconnect from the string**.

   Enter the STRING DISCONNECT directive in the Control window command line:

   STRING DISCONNECT STRING=100

8. **Delete the logical string**.

   Enter the STRING DELETE directive in the command line of the Control window:

   STRING DELETE STRING=*100*

9. **Monitor event messages on the Global Event Display window**.

   Verify the progress of the STRING DELETE request. During string deletion, event messages are generated as processes are terminated. If a process will not terminate, a message stating the String was unsuccessfully terminated with be generated. Repeat step 3 to terminate the active process. Table 9.8-1 identifies the text of event messages associated with string deletion and their significance.

10. **Monitor the parameter updates on the dynamic page**.

    Observe the removal of the logical string from the page.

Table 9.8-1. String Deletion Event Message Text

Event Message Text	Significance
Shutting down __ process.	The process is being shut down.
__ terminated __ Process ID __	The process has been terminated.
String __ was __	Indicates whether the string was successfully terminated.

9.9 Logical String Disconnection

Disconnect from a shared logical string by entering the STRING DISCONNECT directive in the Control window ECL command line and specifying the string ID from which disconnection is desired.

To disconnect from a logical string:

1. **Open the Local Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Disconnect from the string**.

   Enter the STRING DISCONNECT directive in the Control window command line:

   STRING DISCONNECT STRING=*100*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page that includes the logical string from which you wish to disconnect.

4. **Monitor event messages on the Local Event Display window**.

   Verify the progress of the STRING DISCONNECT request. During string disconnection, event messages are generated as processes are terminated. If a process will not terminate, repeat step 2 to terminate the active process. Table 9.9-1 identifies the text of event messages associated with string disconnection and their significance.

5. **Monitor the parameter updates on the dynamic page**.

   Observe the removal of your login name from the list of users connected to the string and the removal of your userstation name from either the Mirrored Userstation List or the Tailored Userstation List, depending on the connection to the string.

Table 9.9-1. String Disconnection Event Message Text

Event Message Text	Significance
Shutting down __ process.	The process is being shut down.
__ terminated __ Process ID __	The process has been terminated.
User __ disconnected from String __	You were successfully disconnected from the string.

9.10 Spacecraft Telemetry Processing

The FOS Telemetry Service receives and processes spacecraft data. All components of the Telemetry Service are initiated as part of a logical string. With respect to processing of spacecraft data, the Telemetry Service provides three distinct capabilities: data decommutation, memory dump, and state check. Decommutation is performed on both real-time and historical telemetry data. The real-time telemetry data is received from EDOS and historical telemetry data is received from the Data Management Service (DMS).

Data are received by Telemetry tasks through a set of well-known ports and addresses. The port and address used is determined by the mode; either operational, test, or training. The actual port and address is retrieved from the NameServer through the use of a service name made up of the mode and spacecraft. If the mode is operational, the data source must be real-time; if the mode is training, the data source must be simulation; and if the mode is test, the data source may be either real-time or simulation.

The memory dump function provides validation, file storage, and subsystem access to real-time spacecraft memory dump data. The spacecraft state check capability allows the FOS to store the current state of the spacecraft, monitor and compare the spacecraft's state with this baseline, and compare the spacecraft's state with its expected state.

ECL directives control the function of the Telemetry Service. Through these directives, parameters may be altered to modify data decommutation and conversion, event message generation, parameter limit processing, and spacecraft state checking.

The following sections provide detailed information about Telemetry subsystem control directives and the three functional components.

9.10.1 Directives

The following sections discuss ECL directives used to control and modify archiving, decommutation, EU conversion, limit checking, and derived parameter processing of telemetry data, and the ECL directive used to initiate spacecraft state checking. Refer to Appendix A for the syntax of ECL directives.

9.10.1.1 ARCHIVE Directive

The ARCHIVE directive controls telemetry data archiving. Modifications made by ARCHIVE directives are in effect until data archiving is redefined or the telemetry software is restarted.

9.10.1.2 DECOM Directive

The DECOM directive controls selective decommutation of telemetry parameters. The decommutation of telemetry parameters may be turned on or off for all parameters, for parameters by subsystem, or for individual parameters. Modifications made by DECOM directives remain in effect until additional modifications are made or the telemetry software is restarted.

9.10.1.3 DERIVED Directive

The DERIVED directive controls the evaluation and update of derived telemetry parameters. The DERIVED directive allows equation processing intervals to be adjusted. Setting the interval to a value of zero disables derived parameter processing. The equation processing rate is based upon spacecraft time extracted from telemetry.

Derived parameter processing involves a processing capability accommodating special computations using predefined algorithms. These simple calculations are defined in the PDB. Predefined analog, discrete, constant, or other derived values may be sources for building a new derived parameter.

9.10.1.4 DROPOUT Directive

The DROPOUT directive controls the sensing rate for detecting the loss of an input telemetry stream. The DROPOUT directive allows the dropout sensing interval to be displayed and adjusted. Setting the interval to a value of zero disables dropout detection.

9.10.1.5 EU Directive

The EU directive controls the coefficient changes of polynomial engineering unit conversion equations for telemetry parameters that have EU conversions defined in the database. The EU directive allows changes to be made to the coefficient values of EU equations and the selection of a conversion algorithm. Changes made to coefficient values for EU equations are in effect until coefficient values are redefined or the telemetry software is restarted.

9.10.1.6 LIMITS Directive

The LIMITS directive controls the limit sensing of parameters, enabling/disabling of limit messages and alarms, frequency of limit messages, and changing of individual limit values.

High/low limit checking and delta limit checking are performed on telemetry points whose parameter specification in the database specify these checks. The limits may be checked in raw or EU values.

For high or low limits checking, the actual value of the telemetry parameter is compared to the red and yellow limits defined for that parameter and reports violations in event messages. A parameter value outside yellow limits indicates that normal operational limits have been violated. A parameter value outside red limits indicates that the parameter has reached a dangerous value and that some action is required. Sets of red and yellow limits are defined in the database, but can be modified in real-time with the LIMITS directive. The selection of a set of red and yellow limits is made according to the current value of a discrete telemetry parameter called a limit switch.

With the LIMITS directive, high/low limit reporting can be turned on or off during a contact for one parameter, for a subsystem's parameters, or for all parameters.

For delta limit checking, the current value of a telemetry parameter is compared with the previous value of that parameter calculating a parameter delta (absolute difference). If the parameter delta exceeds the delta limit, a delta limit violation is reported in an event message. A delta limit can be defined in raw counts in the database and can be changed temporarily with the LIMITS directive. Changes to limits remain in effect until limits are redefined or the telemetry software is restarted.

9.10.1.7 STATE Directive

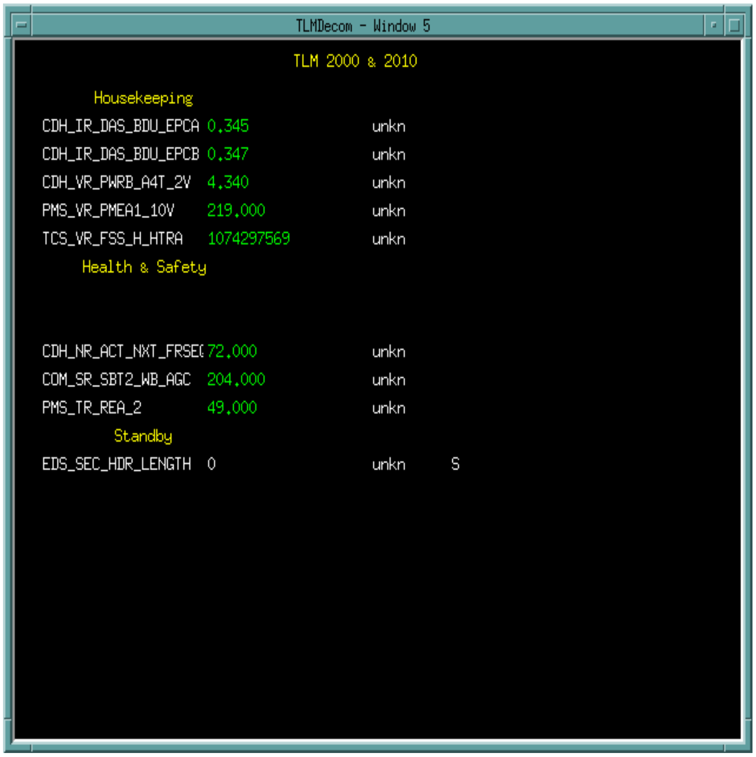
The STATE directive controls the execution of spacecraft state checking and capturing.

The STATE directive allows the selection of a spacecraft table to be used for state checking; the replacement of the expected state table with current telemetry values; the start of the spacecraft state check comparison; and the switching of the telemetry channel to be state checked.

9.10.1.8 Data Decommutation

The Telemetry Service decommutates telemetry data received in EDU packets. Real-time data is received from EDOS via the User Datagram Protocol (UDP) over dedicated, multicast IP addresses and ports assigned per mission as defined in the applicable Operations Agreement (OA). The specific port on which each data type is received from EDOS is based on the Application ID (APID). Historical replay data is received from the DMS through a network point-to-point connection. Three types of telemetry data are received and processed by the Telemetry service: Housekeeping (HK), Health and Safety (HS), and Standby (SB).

Decommutation of telemetry data is based on parameter definitions from the database. These definitions affect the way the Telemetry Service decommutates a parameter. Several operations may be performed on a parameter after it has been decommutated. A converted value may be obtained by using one of three EU conversions: polynomial, exponential, or line segment. Either the decommutated or converted value may be high/low and/or delta limit checked using predefined limit sets. New telemetry parameters or derived parameters may be computed by performing mathematical operations on the value of one or more existing telemetry parameters. Each parameter in each EDU packet is processed according to type and the parameters sent to a Parameter Server for access by other subsystems. The EDU packet may be sent to the DMS for archiving and further analysis. Figure 9.10.1.8-1 is a sample display page containing decommutated telemetry values.



```
                              TLM 2000 & 2010

      Housekeeping
  CDH_IR_DAS_BDU_EPCA 0.345              unkn
  CDH_IR_DAS_BDU_EPCB 0.347              unkn
  CDH_VR_PWRB_A4T_2V  4.340              unkn
  PMS_VR_PMEA1_10V    219.000            unkn
  TCS_VR_FSS_H_HTRA   1074297569         unkn
     Health & Safety


  CDH_NR_ACT_NXT_FRSE(72.000             unkn
  COM_SR_SBT2_WB_AGC  204.000            unkn
  PMS_TR_REA_2        49.000             unkn
        Standby
  EDS_SEC_HDR_LENGTH  0                  unkn    S
```

Figure 9.10.1.8-1. Telemetry Display Page

9.11 Memory Dump

The Telemetry Service provides the capability to receive and store spacecraft memory dump data. This data is received from EDOS via the UDP over dedicated, multicast IP addresses and ports. The Telemetry Service validates memory dump data and saves it in a file. When the dump is complete, the DMS and CMS are notified that the memory dump file is ready for storage and additional processing.

When a spacecraft memory dump begins, the Telemetry Service issues an event message containing the memory dump filename and expected size. Memory dump EDU packets are received by the Telemetry Service at either the 16K or 1K rate, validated according to Remote Terminal ID and word count in the EDU, and saved in the dump file. When the memory dump is complete, the Telemetry Service issues an event message containing the filename and number of words dumped. The DMS and CMS are notified that the dump is complete and the file is ready for archiving.

Three additional event messages may be issued by the Telemetry Service in the course of processing a memory dump. An event is generated if a time-out condition exists while receiving memory dump EDUs; if the memory dump is aborted before all EDUs have been received; or if the Remote Terminal ID differs from the ID expected. In any case, the event message indicates the memory dump is incomplete. After a time-out condition, the Telemetry Service is ready for another memory dump to begin.

9.12 State Check

The spacecraft State Check capability of the Telemetry Service is used to assist in back-orbit command verification. It compares the current state of the spacecraft with those expected by comparing the value of selected telemetry parameters with their expected values.

This component performs one of four functions as directed by a request received from the FOS User Interface (FUI) subsystem. First, a table of expected values for selected parameters may be loaded. These are the values to which real-time parameter values will be compared. This table of expected values will be received by the Telemetry Service from the Command Management Service (CMS) subsystem. The second function baselines selected parameters with values currently in the Parameter server. The third request may direct the telemetry subsystem to perform the state check by comparing the values of parameters in the table of expected values against those retrieved from the Parameter server. Lastly, a request may direct the Telemetry Service to change channels. This allows the Telemetry Service to retrieve parameters from a different Parameter server.

9.13 Ground Telemetry Processing

The EOC processes certain ground telemetry messages to assist in assessing spacecraft contact service sessions provided by the NCC and EDOS. Ground telemetry messages include periodic status information from the NCC and EDOS; real-time Ground Control Message Requests (GCMR) to NCC; and messages aiding EOC spacecraft clock calibration and correlation computations.

9.13.1 Directives

Directives for ground telemetry processing are entered via the command line of the Command Control window. The results of ground telemetry processing are monitored through event messages displayed on the Event Display window and parameter updates on dynamic pages. Directives applicable to ground telemetry processing are derived from the RCCONFIG, GCMR, and NCC keywords. The RCCONFIG directive has associated secondary keywords that configure the RCM subsystem. The GCMR directive has various secondary keywords that send GCMRs to the NCC. The NCC directive contains associated secondary keywords that enable and disable User Performance Data (UPD). It also has a secondary keyword specifying which NCC service is designated to conduct communication tests. The following sections describe in detail the usage of these directives.

9.13.2 Network Control Center User Performance Data

The Ground Telemetry Processing Service allows you to enable and disable UPD from the NCC UPD service via directives. It also provides a directive for conducting communication tests with the NCC UPD service. In order to send a UPD Request and conduct the communication test for the NCC UPD service, you must have proper command authority.

To send a User Performance Data request:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   In the Control window's command line enter the STRING CONNECT directive to establish a connection to the logical string in a mirrored configuration:

   STRING CONNECT STRING=*100* CONFIG=*MIRROR*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page which displays UPD.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is not successful, investigate why by reviewing the event messages generated and repeat step 3 before proceeding.

5. **Assume command authority**.

   Enter the TAKE COMMAND directive in the command line of the Control window:

   TAKE COMMAND STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify the progress of the authority request. If command authority is granted, an event message stating "Command Authority has changed from __ to __" will be generated.

7. **Monitor the parameter updates on the dynamic page**.

   Observe updates to the current command user login identifier and command userstation identifier.

8. **Enter the command enabling NCC UPD in the command line of the Command Control window**.

   NCC UPD *ENABLE*

9. **Monitor the Local Event Display window and dynamic page**.

   Observe event message and parameter updates to verify the UPD Request was sent to the NCC.

To conduct a communication test for NCC UPD Service:

1. **Follow steps 1 through 7 of paragraph 9.13.2, describing how to send a User Performance Data Request**.

2. **Enter the communications test command for NCC UPD in the command line of the Command Control window**:

   NCC COMMTEST *UPD*

3. **Monitor the Global Event Display window and dynamic page**.

   Observe event message and parameter updates to verify the communication test message was sent to the NCC UPD Service. Confirm that the system received the Communication Test Message echo from the NCC UPD Service.

9.13.3 Network Control Center Ground Control Message Requests

The Ground Telemetry Processing Service provides directives to send GCMRs to the NCC Reconfiguration Service, as well as a directive to select a Tracking and Data Relay Satellite (TDRS). In order to send a GCMR to the NCC Reconfiguration Service, you must have the proper command authority; to select a TDRS, you need ground control authority (refer to paragraph 9.7).

To select a TDRS:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Local Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   In the Control window's command line enter the STRING CONNECT directive to establish a connection to the logical string in a mirrored configuration:

   STRING CONNECT STRING=*100* CONFIG=*MIRROR*

3. **Open a dynamic page**.

   On the Control window, click **TlmWins...** and select a dynamic page displaying TDRS.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is unsuccessful, investigate why by reviewing the event messages generated and repeat step 3 before proceeding.

5. **Assume ground control privilege**.

   Enter the TAKE GROUNDCONTROL directive in the command line of the Command Control window:

   TAKE GROUNDCONTROL STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify the progress of the authority request. If the TAKE GROUND CONTROL directive is unsuccessful, review event messages generated to determine why you did not take control and repeat step 3 before proceeding.

7. **Monitor the parameter updates on the dynamic page**.

   Observe parameter updates to the dynamic pages.

8. **Enter the RCCONFIG directive in the command line of the Command Control window**.

   RCCONFIG STRING=*100* TDRS=*TDE*

9. **Monitor events on the Local Event Display window and parameters on the dynamic pages**.

   Observe event messages to verify that the TDRS has been reconfigured. Monitor the TDRS parameter update on the dynamic page.

Ground Control Message Requests:

There are six GCMRs you can send to NCC:

1. User Reacquisition Request Message.
2. User Reconfiguration Request Message.
3. Forward Link Sweep Request Message.
4. Forward Link EIRP Reconfiguration Request Message.
5. Expanded User Frequency Uncertainty Request Message.
6. Doppler Compensation Inhibit Request Message.

The FOS provides five different User Reconfiguration Request Messages to request a reconfiguration of specified services (refer to Appendix A for additional details):

1. MA Forward Link Reconfiguration Request.
2. SSA Forward Link Reconfiguration Request.
3. MA Return Link Reconfiguration Request.
4. SSA Return Link Reconfiguration Request.
5. KSA Return Link Reconfiguration Request.

To send a GCMR:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Global Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   In the Control window's command line enter the STRING CONNECT directive to establish a connection to the logical string in a mirrored configuration:

   STRING CONNECT STRING=*100* CONFIG=*MIRROR*

3. **Open dynamic pages**.

   Click **TlmWins...** on the Command Control window and select dynamic page displaying GCMR and NCC parameters.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is unsuccessful, investigate why by reviewing the event messages generated and repeat step 2 before proceeding.

5. **Assume command authority**.

   Enter the TAKE COMMAND directive in the command line of the Command Control window:

   TAKE COMMAND STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify the progress of the authority request. If command authority is granted, an event message stating "Command Authority has changed from __ to __" will be generated.

7. **Monitor the parameter updates on the dynamic pages**.

   Observe parameter updates to the dynamic pages.

8. **Send selected GCMRs to the NCC Reconfiguration Service**.

   Enter the following directive in the command line of the Command Control window:

   GCMR REACQUISITION LINK=*MA_FWD* SUPPORT=*FORWARD*

9. **Monitor events on the Global Event Display window and parameters on the dynamic pages**.

   Observe events on the Event Display to verify that the system received the GCMR Disposition Message and GCMR Status Message from the NCC Reconfiguration Service. Observe parameter updates on the dynamic pages.

To conduct a communication test for NCC Reconfiguration Service:

1. **Follow steps 1 through 7 of paragraph 9.13.3, illustrating how to send a GMCR**.
2. **Enter the following directive in the command line of the Command Control window**.

   NCC COMMTEST *GCMR*

3. **Monitor events on the Global Event Display window and parameters on the dynamic pages**.

   Observe events on the Event Display to verify the system received the Communication Test Message echo from the NCC Reconfiguration Service.

9.13.4 EOS Data and Operations System (EDOS) CODAs

The Ground Telemetry Processing Service provides you with information describing the operational activities of EDOS by spacecraft. EDOS sends the CODA report to the EOC every 5 seconds during a spacecraft contact session. Each report contains updated and accumulated statistics from the beginning of the spacecraft contact session to its end. Updated values can be verified on a user-defined dynamic page displaying CODAs. All CODA reports are archived in a central repository.

9.13.5 EOS Data and Operations System (EDOS) Communication Test

The Ground Telemetry Processing Service provides you with a directive for conducting communication tests with the EDOS Service. You are allowed to reconfigure the timeout value for receipt of the Command Echo Blocks from the EDOS Service.

To conduct a communication test for EDOS Service:

1. **Open the Global Event Display window**.

   Click **Tools…** on the Control window or Command Control window. Select the **Global Event Display** from the list of tools and click **OK**.

2. **Connect to a logical string**.

   In the Control window's command line enter the STRING CONNECT directive to establish a connection to the logical string in a mirrored configuration:

   STRING CONNECT STRING=*100* CONFIG=*MIRROR*

3. **Open a dynamic page**.

   Click **TlmWins...** on the Command Control window and select a dynamic page displaying EDOS parameters.

4. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify that the connection to the desired logical string was successful. If connection is unsuccessful, investigate why by reviewing the event messages generated and repeat step 2 before proceeding.

5. **Assume command authority**.

   Enter the TAKE COMMAND directive in the command line of the Command Control window:

   TAKE COMMAND STRING=*100*

6. **Monitor event messages on the Global Event Display window**.

   Observe event messages to verify the progress of the authority request. If command authority is granted, an event message stating "Command Authority has changed from __ to __" will be generated.

7. **Monitor the parameter updates on the dynamic page**.

   Observe parameter updates to the dynamic pages.

8. **Enter the EDOS COMMTEST directive**.

   EDOS COMMTEST TIMEOUT=*20*

The value 20 is an integer in seconds. The TIMEOUT subdirective is optional. If a TIMEOUT value is not specified, a default value of 8 seconds is used.

9. **Monitor events on the Global Event Display window and parameters on the dynamic page**.

   Observe events on the Event Display window to verify that a communication test message was sent to EDOS and that the system received the command echo block from EDOS.

9.14 Command Processing

The ECS Command Service provides command formatting and metering to the EDOS interface for transmission to the spacecraft. Before a command is formatted, it is validated according to the corresponding database command definition. The following items are checked as part of the validation process: command mnemonic, submnemonic(s), prerequisite telemetry values (if prerequisite checking is enabled; refer to paragraph 9.12), and criticality. In addition, before each command is sent, its binary is compared against a list of hazardous commands maintained in the database. If a hazardous command is found, processing is immediately aborted. Once a command is validated, it is formatted according to Consultative Committee for Space Data Systems (CCSDS) protocols. All commands transmitted are archived and receipt verified. Command execution verification is then performed within the database defined timeout interval.

Commands originate from real-time command directives, loads prepared by the Command Management Service and executed via the LOAD directive, and directives in EOC automated ground scripts issued from the Command Control window.

Command data blocks are transmitted to EDOS using a set of multicast ports and addresses taken from the NameServer via service names based on the operational mode and data source. If the mode is operational, the data source is limited to real-time and the Ops port and address will be used. For the test mode, the port address will be used if the data source is real-time and the training port address will be used if the data source is simulation. When the mode is training, the data source must be simulation and the training port address will be used.

Event messages are used to indicate the processing status of all commands issued to this subsystem. If a command cannot be processed without overriding submnemonic and prerequisite failures or criticality, an event message is generated and the system prompts you to allow or cancel the command. Refer to Appendix B for a list of FOS event messages.

9.14.1 Command Directives

The directives processed by the Command Service are listed and briefly described in the following text. All command directives are entered in the command line of the Command Control window. The **Allow** and **Cancel** buttons on the Command Control window are used to allow or cancel commands whose validation fails because of submnemonic, prerequisite or critical checking.

a. **CMD**. A request to format and send a real-time command from a mnemonic and optional submnemonic(s).

b. **RAW**. Request to format and send a real-time command in binary (ASCII hexadecimal) format.

c. **LOAD**. Initiate a memory load with a load identifier, partition indicator, and load type.

d. **STORED**. Request for a stored command to be execution-verified.

e. **FOP**. Configure the Frame Operation Procedure (FOP) on the ground. Refer to paragraph 9.14.4 for additional details.

f. **CMDCFG**. Set command configuration parameters to select or control prerequisite checking, PLOP 1 or 2, CTIU 1 or 2, and uplink bit rate.

9.14.2 Real-Time Command Processing

Command directives sending real-time commands are entered in the Command Control window manually or received from the ground script controller via ground scripts or procedures. Before a command is validated, the command source is checked to verify that the operator has current commanding authorization. The command mnemonic and submnemonic(s) are compared with the database command definition loaded during initialization. If the submnemonic(s) had no default value or was out of range, the system prompts you to override the failure or cancel the command by clicking the **Override** or **Cancel** Command Control window button.

If prerequisite checking is enabled, multiple database-defined telemetry points are compared with their expected values. Again, if this check fails, the system prompts you to override the failure or cancel command processing by clicking the **Override** or **Cancel** Command Control window buttons. Prerequisite state checking can be enabled or disabled via the CMDCFG directive or by clicking the **Enable** or **Disable** Command Control window button. The command is then checked for criticality. If the command is defined as critical in the database, you will be prompted to allow or cancel it prior to execution via the **Allow** or **Cancel** Command Control window buttons.

A final check is performed after the 1553b bus command is built against the binary hazardous command list (as defined in the database). If the first two data words of the command match any of those in the hazardous command list, the command is immediately aborted. If not found, the command is further formatted in a CCSDS command transfer frame and metered to the EDOS interface at a user-selected rate. Receipt and execution verification is then performed for the command as described in the following sections.

9.14.3 Load Processing

Load files containing CCSDS command packets are built by the CMS and accessed via the DMS. The LOAD directive is used to initiate load processing. Loads may be divided into 4K partitions that may up uplinked at different times. The load is validated and checked for criticality. If the load contains one or more critical commands, the system prompts you to allow or cancel it via the **Allow** or **Cancel** Command Control window button.

Validation includes checking for correct spacecraft ID and that all prior partitions for this load have been uplinked. If a partition is out of order, the system prompts you to allow or cancel it by clicking the **Allow** or **Cancel** button. Prerequisite checking may also be performed against the command contained in the load that initiated the memory load onboard the spacecraft.

Receipt verification is performed for each packet in the load. When the last packet is receipt-verified, end-item telemetry verification is performed to ensure that the Cyclic Redundancy Check (CRC) for the load was successful.

ABORT may be selected to terminate the uplink of a load in progress.

Command directives to send real-time commands are entered in the Command Control window manually or received from the ground script controller via ground scripts or procedures. Before a command is validated, the command source is checked to verify that the operator has current commanding authorization. The command mnemonic and submnemonic(s) are compared with the database command definition loaded during initialization. If the submnemonic(s) are not assigned a value or if the value is out of range, the system prompts you to override the failure or cancel the command by clicking the **Override** or **Cancel** Command Control window button.

If prerequisite checking is enabled, multiple database-defined telemetry points are compared with their expected values. Prerequisite state checking can be enabled or disabled via the CMDCFG directive or via the **Enable** and **Disable** Command Control window buttons. Again, if this check fails, the system prompts you to override the failure or cancel the command by clicking the **Override** or **Cancel** button on the Command Control window. The command is then checked for criticality as defined in the database. Critical commands must be either allowed or canceled prior to execution by clicking the **Allow** or **Cancel** Command Control window buttons.

A final check is performed after the 1553b bus command is built against the binary hazardous command list, as defined in the database. If the first two data words of the command match any of those in the hazardous command list, the command is immediately aborted. If not found, the command is further formatted in a CCSDS command transfer frame and metered to the EDOS interface at a user-selected rate. Receipt and execution verification is then performed for the command as previously described.

9.14.4 FOP Control and Processing

The FOP provided by the Command Service permits you to keep the EOC in synchronization with the Frame Acceptance and Reporting Mechanism (FARM) on the spacecraft. The FOP and FARM together comprise the Command Operation Procedure (COP, COP1 for AM-1) and ensures and verifies that telecommand transfer frames are received by the spacecraft without omission or duplication, and in the same sequential order in which they were transmitted. For AM-1 a transfer frame contains a single packet containing a single 1553b command. For a complete discussion of COP1 and the CCSDS protocol, refer to CCSDS documentation.

The two basic elements of COP1 are the transfer frame sequence number and the Command Link Control Word (CLCW) downlinked by the spacecraft. The spacecraft expects transfer frames to be ordered sequentially via a modulo 256 frame sequence number. FOP assigns sequential sequence numbers to each (type AD) transfer frame transmitted. The CLCW contains the spacecraft's "next expected frame sequence number" (VR), from which the "last accepted frame sequence number" may be computed. This is used to initially synchronize the ground with the spacecraft and to verify that frames have been received. The CLCW also contains error flags to indicate that retransmission is needed; a wait is needed (should be prevented from ever occurring by command metering); or a lockout (sequence numbers out of order to an extreme degree) has occurred.

Before any commands can be sent, the FOP must be initialized and in the "Active" (ready to command) state. This implies that the ground and the spacecraft are known or assumed to be in synch with each other. This can be accomplished by synchronizing the ground with the spacecraft (the norm), or the spacecraft with the ground. The FOP directive cases used to initialize the FOP are:

a. **FOP INIT CHECK**. This synchronizes the ground to the spacecraft by receiving a CLCW and using its VR for the next frame to be sent sequence number. If a clean CLCW (no error flags set) is not received before the selected timer expires, initialization will be considered unsuccessful.

b. **FOP INIT VR=<*next expected frame sequence number*>**. This synchronizes the spacecraft to the ground by sending a type BC control command (set VR) to set its VR value to the requested value. Control commands are not subjected to the frame sequence number check performed for regular (type AD) commands. After the set VR command is sent, a clean CLCW is expected with a VR value identical to the one entered in the directive.

c. **FOP INIT NOCHECK**. Do not wait for a CLCW, start commanding with whatever FOP's next frame sequence number to be sent is set to. This would usually be done when telemetry, and hence CLCWs, are unavailable ("blind commanding"). A typical scenario would be to disable retransmission, set the time out to a small value, enter FOP VS=<n> directive to tell FOP what sequence number to use next, then FOP INIT VR=<n> to send a set VR command (knowing that it will not be acknowledged by a CLCW but will time out and hence finish quickly), and finally, FOP INIT NOCHECK. The queue size and the time out should then be set to the maximum allowed values to allow as many commands to be sent as possible before a time out occurs and the FOP goes into retransmission mode. Then, the above scenario would have to be repeated.

d. **FOP INIT UNLOCK**. This would only be used when the CLCW lockout flag is set. It sends a BC Unlock command to the spacecraft and then waits for a clean CLCW, leaving the FOP in the "Active" state.

The FOP is always in one of six states, as indicated on the Command Parameters dynamic page (see Figure 9.14.4-1). They are:

a. **Initial**. FOP has not been initialized, commanding not permitted. This would be expected after FOS has been brought up but no FOP directives entered.

b. **InitializingWithBCFrame**. Type BC frame sent (unlock or set VR) but clean CLCW not yet received.

c. **InitializingWithoutBCFrame**. FOP INIT CHECK entered and waiting for clean CLCW.

d. **RetransmitWithWait**. A CLCW has been received with the retransmit and wait flags set. The FOP is in retransmission mode and no new commands are permitted.

e. **RetransmitWithoutWait**. A CLCW has been received with the retransmit flag set. The FOP is in retransmission mode and no new commands are permitted.

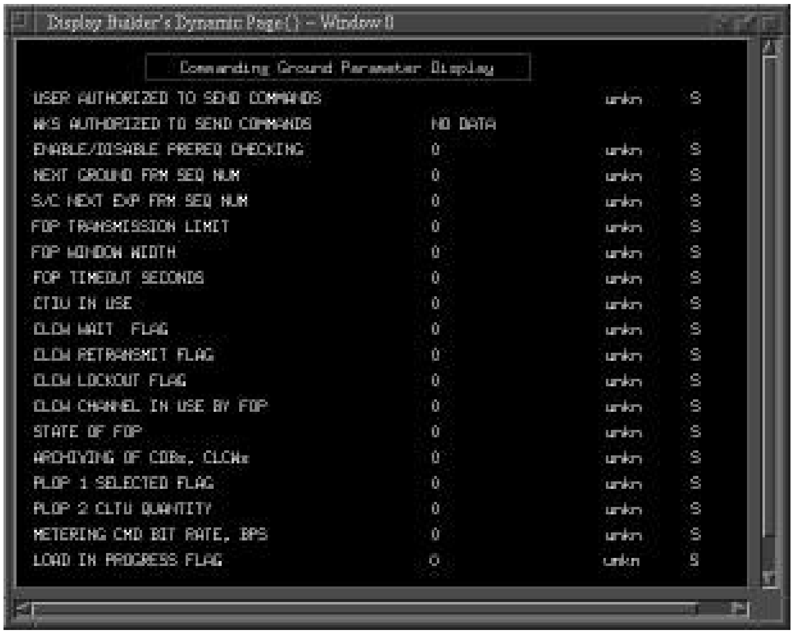
f. **Active**. FOP is initialized and ready to send commands or commands have been sent and are pending verification but no problems have occurred.

As FOP sends frames, it saves them in a queue awaiting CLCW verification. The size of the queue is referred to as the "sliding window width" and is adjustable via the FOP directive (FOP WINDOW=<window width>). As CLCWs arrive, an incrementing VR indicates frames verified and removes them from the queue. If a frame is received by the spacecraft with an out of order sequence number, it is rejected and the "retransmit flag" bit is set in the CLCW. This causes FOP to retransmit all the frames in its queue and disable transmission of new frames. The number of transmissions is also controlled by the FOP directive (FOP TRANSMIT=<transmissionLimit>).

It sets the total number of transmissions allowed, so setting it to one disables retransmission. In order to keep from declaring that the link has failed when it is in fact getting frames into the spacecraft, the transmission count limit applies only to the first (oldest) frame in the queue. Once that frame is acknowledged, the count is reset, even though the remaining frames on the queue have been transmitted, possibly more than once. The effect is that the transmission count can be considered to be associated with the queue, rather than with each frame.

As FOP sends frames, it saves them in a queue awaiting CLCW verification. The size of the queue is referred to as the "sliding window width" and is adjustable via the FOP directive (FOP WINDOW=<window width>). As CLCWs arrive, an incrementing VR indicates frames verified and removes them from the queue. If a frame is received by the spacecraft with an out of order sequence number, it is rejected and the "retransmit flag" bit is set in the CLCW. This causes FOP to retransmit all the frames in its queue and disable transmission of new frames. The number of transmissions is also controlled by the FOP directive (FOP TRANSMIT=<transmissionLimit>).

It sets the total number of transmissions allowed, so setting it to one disables retransmission. In order to keep from declaring that the link has failed when it is in fact getting frames into the spacecraft, the transmission count limit applies only to the first (oldest) frame in the queue. Once that frame is acknowledged, the count is reset, even though the remaining frames on the queue have been transmitted, possibly more than once. The effect is that the transmission count can be considered to be associated with the queue, rather than with each frame.



Commanding Ground Parameter Display

USER AUTHORIZED TO SEND COMMANDS		unkn	S
WKS AUTHORIZED TO SEND COMMANDS	NO DATA		
ENABLE/DISABLE PREREQ CHECKING	0	unkn	S
NEXT GROUND FRM SEQ NUM	0	unkn	S
S/C NEXT EXP FRM SEQ NUM	0	unkn	S
FOP TRANSMISSION LIMIT	0	unkn	S
FOP WINDOW WIDTH	0	unkn	S
FOP TIMEOUT SECONDS	0	unkn	S
CTIU IN USE	0	unkn	S
CLCW WAIT FLAG	0	unkn	S
CLCW RETRANSMIT FLAG	0	unkn	S
CLCW LOCKOUT FLAG	0	unkn	S
CLCW CHANNEL IN USE BY FOP	0	unkn	S
STATE OF FOP	0	unkn	S
ARCHIVING OF CDBs, CLCWs	0	unkn	S
PLOP 1 SELECTED FLAG	0	unkn	S
PLOP 2 CLTU QUANTITY	0	unkn	S
METERING CMD BIT RATE, BPS	0	unkn	S
LOAD IN PROGRESS FLAG	0	unkn	S

Figure 9.14.4-1. Command Parameters Display

FOP also expects to get CLCWs periodically and if it does not receive one within a time limit selected by the user via the FOP directive (FOP TIMER=<timeoutValueInSeconds>), it will begin retransmission, if enabled by the user and if frames are in the verification queue. The timeout value is also used while initializing the FOP and waiting for CLCWs, as previously discussed.

The spacecraft has two command and telemetry interface units (CTIU) to which commands may be directed. Each CTIU has its own CLCW, referred to as “I” and “Q” channel. Normally the Q channel CLCW is associated with CTIU-1 but is changeable via commands to the spacecraft and the ground system configuration at EDOS and WSGT. The destination CTIU and the CLCW channel to be used for frame acceptance verification are selected via the CMDCFG directive.

9.14.5 PLOP1/PLOP2

The Physical Layer Operation Procedure (PLOP) is provided by the Command Service. The CMDCFG directive is used to select PLOP1 or PLOP2. For PLOP1 a Command Data Block (CDB) to be sent to EDOS always contains a single Command Link Transmission Unit (CLTU) followed by an 8 byte gap sequence of all zeroes.

PLOP2 for real-time commands is identical to PLOP1 except that there is no gap at the end.

When PLOP2 is selected and a load is being uplinked, the number of CLTUs to be placed in a CDB is determined by the CLTU quantity selected when PLOP2 was selected. For example, if the CLTU quantity selected was 30, 30 CLTUs would be accumulated, placed in a CDB, and sent to EDOS. The detection of the end of the load before 30 was reached would cause the CDB to go out immediately, regardless of the number of CLTUs placed in the CDB thus far.

WARNING

> If the FOP sliding window size is set to less than the PLOP2 CLTU quantity when a load is being uplinked in PLOP2, the Command Service (unless the load is small) could be hung! This is because the FOP queue will be full with FOP waiting for verification status on frames sent and will not forward any more CLTUs to the PLOP portion of the Command Service. PLOP will be waiting for more CLTUs to place in a CDB because it has not yet accumulated the desired amount or detected the end of the load. If this occurs, you will have to wait for FOP to time out and “retransmit” the frames (even though they have not yet been transmitted). This will result in frame sequence errors, etc., and the FOP will probably have to be reinitialized.

9.14.6 Command Failover Processing

When Command Service software is initialized it checks to see if it is “active” or “backup.” If active, it checkpoints its configuration state (parameters that may be set via the CMDCFG or FOP directives) and thereafter whenever the state is changed. If backup, it retrieves the active process configuration state and then receives all directives in parallel with the active service. If and when the backup becomes active, it will be in synchronous regarding configuration state and probably synchronous regarding dynamic state (verification queues, etc.). You are urged to bring up a backup string before any commanding is done on the active string to ensure that the two strings will be synchronous should a failover be necessary.

9.14.7 Execution Verification

When commands are defined in the database, a telemetry parameter may be associated with the command and a value it is expected to take as a result of the command being executed. The length of time allowed for verification (time out value in seconds) is also defined in the database. When a command is sent, it is added to a queue of commands awaiting execution verification that commences as soon as the command is metered to EDOS. The queue is checked by a timer set for the minimum execution verification timeout of the queue. An event message for each command will be generated indicating a successful verification or timeout, at which time it is removed from the queue.